REQUIREMENTS SPECIFICATION

for

ECCD DBMS

Prepared by : 12190011
12190004
12190022

Submitted to : Mulualem Teku
Lecturer

December 28, 2020

1 Requirement Gathering

1.1 Introduction/Background

Early Childhood Care and Education is known in Bhutan as Early Childhood Care and Development, ECCD.
The ECCD, here at Gyelpozhing is known as Kurichuu Early Childhood Care and Development. It is a coorperate ECCD started by the Kurichhu project in August 1, 2010. The ECCD was mainly established for the wellbeing for the children of the employees working in Kurichuu project.

1.2 Vision and Mission

Kurichuu Early Childhood Care and Development, vision and mission:
"Learn and Play"

1.3 Scope

The client for our DBMS project we will be Kurichuu Early Childhood Care and Development, in Gyelpozhing, Mongar.

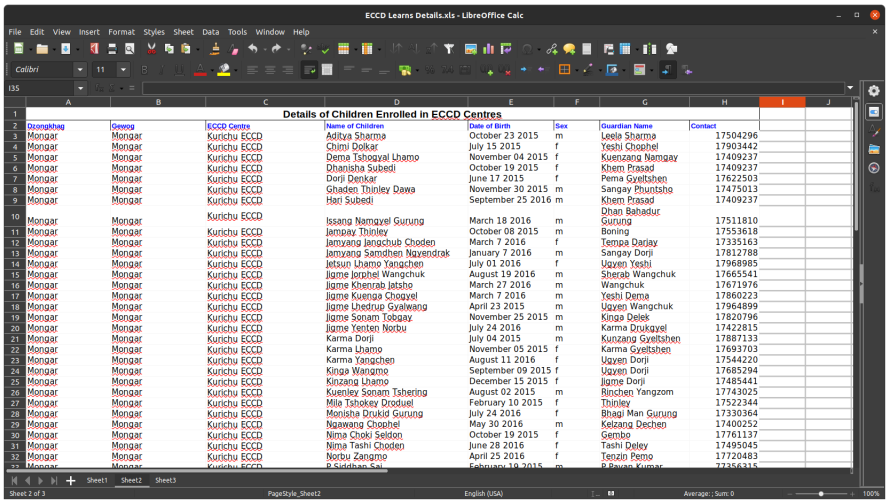
1.4 Objective

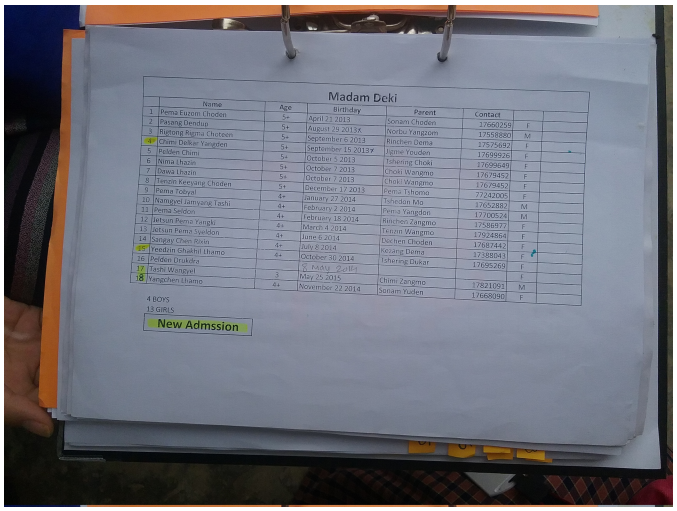
With this project we will be able to design a database managent system suitable for the ECCD

1.5 Study of Similar Systems

- Paper-based File System
  System of storing data in paper files, this system started eversince paper was invented. There are small organisations who follow the paper-based file system even today.
  However it is widely held that keeping and maintaining paper-based documents is unnecessarily cumbersome and costly, now that we have the option of maintaining electronic versions of documents.

- Spreadsheet
  The Kurichhu ECCD, follows a paper based file system. Mainly because the organisation is small in number, with four facilitator and 54 children.
  The information regarding the children are maintained in paper-based files as well as in spreed-sheet.



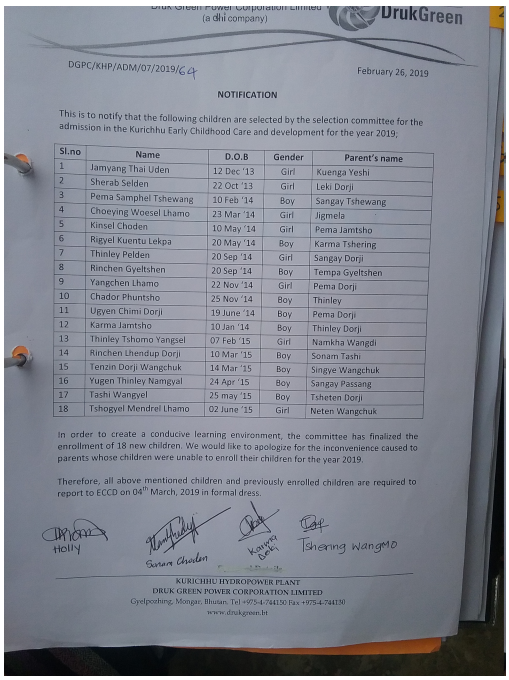


Madam Deki

	Name	Age	Birthday	Parent	Contact	
1	Pema Euzom Choden	5+	April 23 2013	Sonam Choden	17660259	F
2	Pasang Dendup	5+	August 29 2013	Norbu Yangzom	17558880	M
3	Rigtong Rigma Choteen	5+	September 6 2013	Rinchen Dema	17575492	F
4	Chimi Dekar Yangden	5+	September 15 2013	Jigme Youden	17699926	F
5	Pelden Chimi	5+	October 5 2013	Tshering Choki	17699849	F
6	Nima Lhazin	5+	October 7 2013	Choki Wangmo	17679452	F
7	Dawa Lhazin	5+	October 7 2013	Pema Tshomo	77242005	F
8	Tenzin Keeyang Choden	5+	December 17 2013	Tshedun Mo	17652882	M
9	Pema Tobyal	4+	January 27 2014	Pema Yangden	17700524	M
10	Namgyel Jamyang Tashi	4+	February 2 2014	Rinchen Zangmo	17586977	F
11	Pema Seldon	4+	February 18 2014	Tenzin Wangmo	17924864	F
12	Jetsun Pema Yangki	4+	March 4 2014	Dechen Choden	17687442	F
13	Jetsun Pema Syeldon	4+	June 6 2014	Kezang Dema	17388043	F
14	Sangay Chen Rixin	4+	July 8 2014	Tshering Dukar	17695269	F
15	Yeedzin Ghakhil Lhamo	4+	October 30 2014			F
16	Pelden Drukdra	3	May 25 2015	Chimi Zangmo	17821051	M
17	Tashi Wangyel	4+	November 22 2014	Sonam Yuden	17668090	F
18	Yangchen Lhamo					

4 BOYS
13 GIRLS
New Admssion

For each year they maintain a different file.



Druk Green Power Corporation Limited
(a DHI company)

DrukGreen

DGPC/KHP/ADM/07/2019/64

February 26, 2019

NOTIFICATION

This is to notify that the following children are selected by the selection committee for the admission in the Kurichhu Early Childhood Care and development for the year 2019;

Sl.no	Name	D.O.B	Gender	Parent's name
1	Jamyang Thai Uden	12 Dec '13	Girl	Kuenga Yeshi
2	Sherab Selden	22 Oct '13	Girl	Leki Dorji
3	Pema Samphel Tshewang	10 Feb '14	Boy	Sangay Tshewang
4	Choeying Woesel Lhamo	23 Mar '14	Girl	Jigmela
5	Kinsel Choden	10 May '14	Girl	Pema Jamtsho
6	Rigyel Kuentu Lekpa	20 May '14	Boy	Karma Tshering
7	Thinley Pelden	20 Sep '14	Girl	Sangay Dorji
8	Rinchen Gyeltshen	20 Sep '14	Boy	Tempa Gyeltshen
9	Yangchen Lhamo	22 Nov '14	Girl	Pema Dorji
10	Chador Phuntsho	25 Nov '14	Boy	Thinley
11	Ugyen Chimi Dorji	19 June '14	Boy	Pema Dorji
12	Karma Jamtsho	10 Jan '14	Boy	Thinley Dorji
13	Thinley Tshomo Yangsel	07 Feb '15	Girl	Namkha Wangdi
14	Rinchen Lhendup Dorji	10 Mar '15	Boy	Sonam Tashi
15	Tenzin Dorji Wangchuk	14 Mar '15	Boy	Singye Wangchuk
16	Yugen Thinley Namgyal	24 Apr '15	Boy	Sangay Passang
17	Tashi Wangyel	25 may '15	Boy	Tsheten Dorji
18	Tshogyel Mendrel Lhamo	02 June '15	Girl	Neten Wangchuk

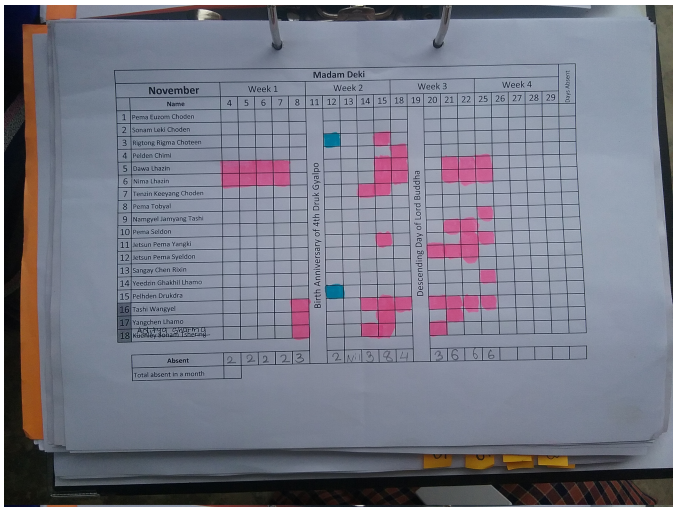
In order to create a conducive learning environment, the committee has finalized the enrollment of 18 new children. We would like to apologize for the inconvenience caused to parents whose children were unable to enroll their children for the year 2019.

Therefore, all above mentioned children and previously enrolled children are required to report to ECCD on 04th March, 2019 in formal dress.

Holly Sonam Choden Kezang Deki Tshering Wangmo

KURICHHU HYDROPOWER PLANT
DRUK GREEN POWER CORPORATION LIMITED
Gyelpozhing, Mongar, Bhutan. Tel +975-4-744130 Fax +975-4-744130
www.drukgreen.bt

The record for attendance is also maintained in the same file.



Madam Deki

November

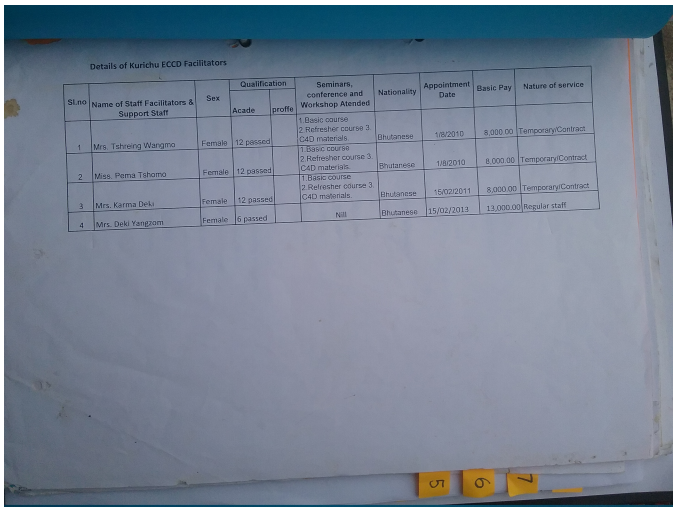
	Name	Week 1					Week 2					Week 3					Week 4					Sick Absent
		4	5	6	7	8	11	12	13	14	15	18	19	20	21	22	25	26	27	28	29	
1	Pema Euzom Choden																					
2	Sonam Leki Choden																					
3	Rigtong Rigma Choteen																					
4	Pelden Chimi																					
5	Dawa Lhazin																					
6	Nima Lhazin																					
7	Tenzin Keeyang Choden																					
8	Pema Tobyal																					
9	Namgyel Jamyang Tashi																					
10	Pema Seldon																					
11	Jetsun Pema Yangki																					
12	Jetsun Pema Syeldon																					
13	Sangay Chen Rixin																					
14	Yeedzin Ghakhil Lhamo																					
15	Pelden Drukdra																					
16	Tashi Wangyel																					
17	Yangchen Lhamo																					
18	Kinsel Choden Tshering																					

Birth Anniversary of 4th Druk Gyalpo

Descending Day of Lord Buddha

Absent	2	2	2	2	3	2	W	3	3	4	3	6	6	6							
Total absent in a month																					

The information on the facilitator are kept in the paper file.



Details of Kurichu ECCD Facilitators

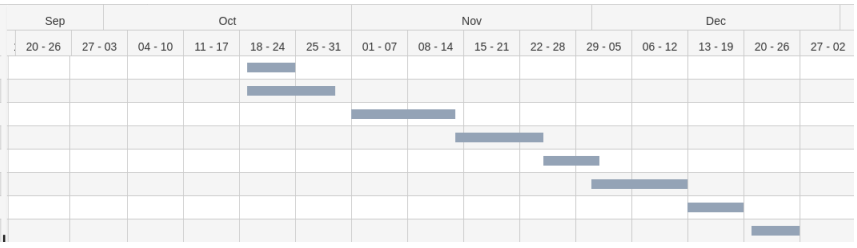
Sl.no	Name of Staff Facilitators & Support Staff	Sex	Qualification		Seminars, conference and Workshop Attended	Nationality	Appointment Date	Basic Pay	Nature of service
			Acade	profe					
1	Mrs. Tshering Wangmo	Female	12 passed		1.Basic course 2.Refresher course 3. C4D materials.	Bhutanese	1/8/2010	8,000.00	Temporary/Contract
2	Miss. Pema Tshomo	Female	12 passed		1.Basic course 2.Refresher course 3. C4D materials.	Bhutanese	1/8/2010	8,000.00	Temporary/Contract
3	Mrs. Karma Deki	Female	12 passed		1.Basic course 2.Refresher course 3. C4D materials.	Bhutanese	15/02/2011	8,000.00	Temporary/Contract
4	Mrs. Deki Yangzom	Female	6 passed		Nill	Bhutanese	15/02/2013	13,000.00	Regular staff

1.6 Expected ER Model

- Entities
  - Facilitator
  - Children
  - Guardian
  - Workshop
- Relationship among the Entities
  - Facilitator cares Children
  - Child has a Guardian
  - Facilitator attends Workahop

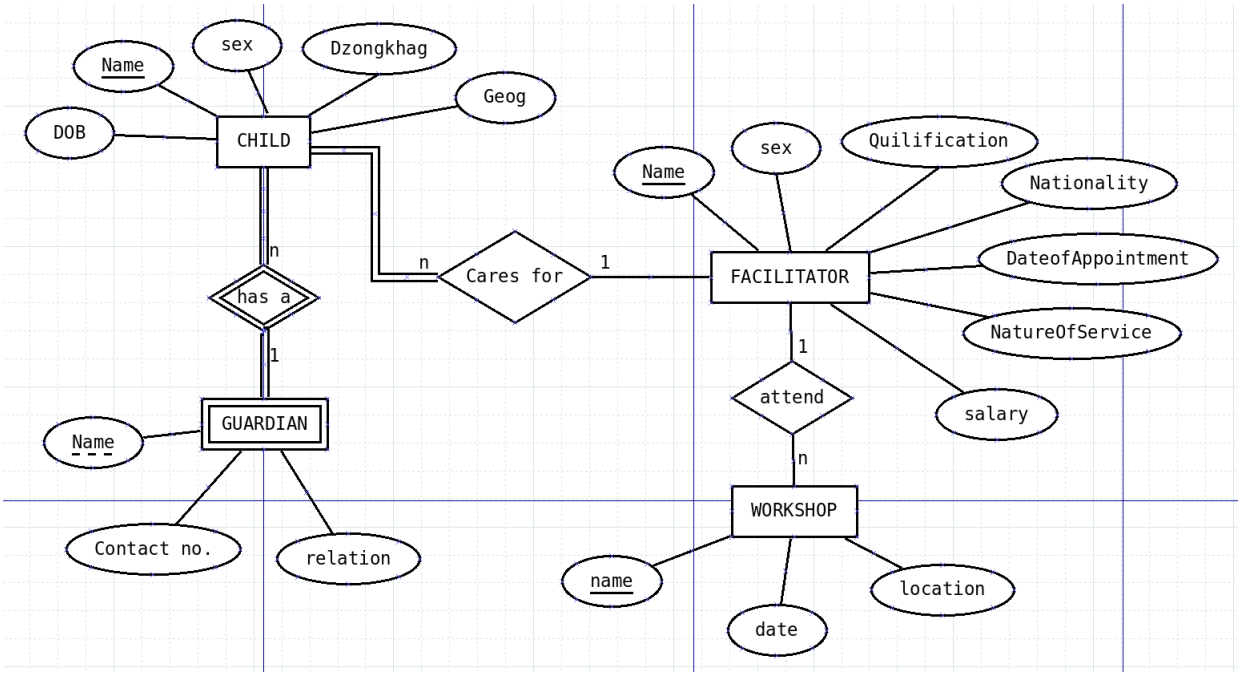
1.7 Work Plan

ID	Title	Start Time	End Time
6	Data collection on client	10/19/2020	10/25/2020
1	Semester Project(Progress check 1)	10/19/2020	10/30/2020
5	ER diagram For Client	11/01/2020	11/14/2020
4	Design DBMS	11/14/2020	11/25/2020
7	Implement and loading	11/25/2020	12/02/2020
3	Semester Project(Progress check 2)	12/01/2020	12/13/2020
8	GUI with tkinter, fine tuning	12/13/2020	12/20/2020
2	Semester Project(FInal Presentation)	12/21/2020	12/27/2020

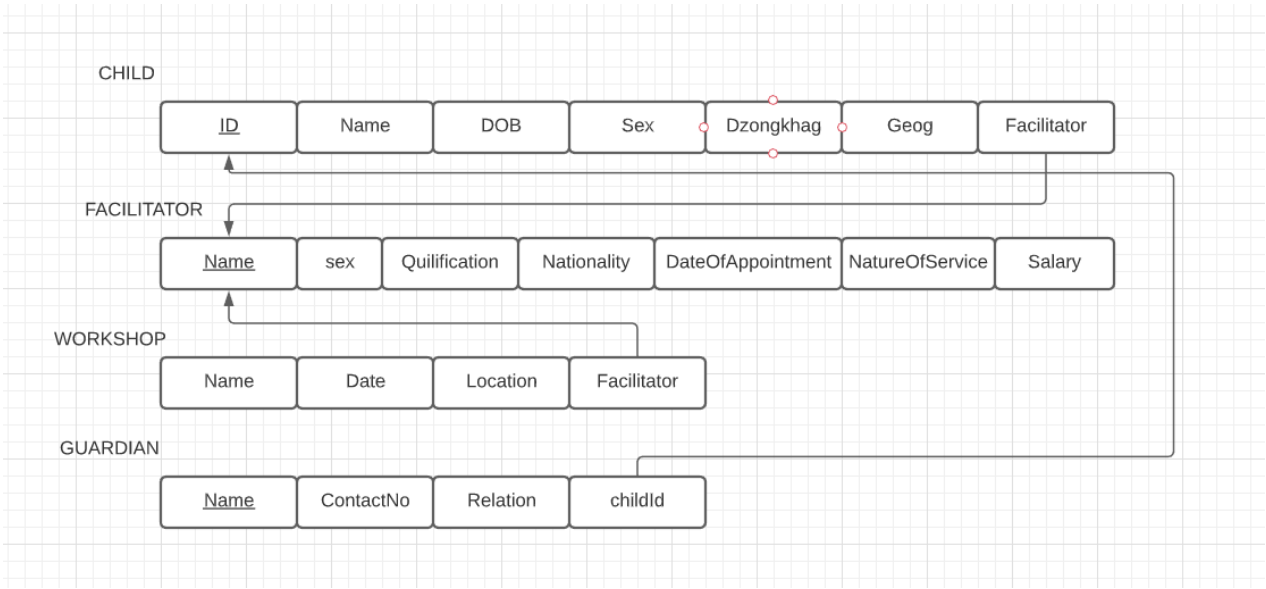


2 Database Design

2.1 ER model of ECCD



2.2 Logical Design



3 Implementation

3.1 Physical Design

```
mysql> describe CHILD;
+-----------------+--------------+------+-----+---------+----------------+
| Field           | Type         | Null | Key | Default | Extra          |
+-----------------+--------------+------+-----+---------+----------------+
| Id              | int unsigned | NO   | PRI | NULL    | auto_increment |
| name            | varchar(30)  | YES  |     | NULL    |                |
| sex             | char(1)      | YES  |     | NULL    |                |
| DOB             | date         | YES  |     | NULL    |                |
| Dzongkhag       | varchar(20)  | YES  |     | NULL    |                |
| Geog            | varchar(20)  | YES  |     | NULL    |                |
| FacilitatorName | varchar(30)  | YES  | MUL | NULL    |                |
+-----------------+--------------+------+-----+---------+----------------+
7 rows in set (0.02 sec)

mysql> describe GUARDIAN;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| Name       | varchar(30)  | NO   | PRI | NULL    |       |
| Contact_No | int          | YES  |     | NULL    |       |
| Relation   | varchar(20)  | YES  |     | NULL    |       |
| childId    | int unsigned | YES  | MUL | NULL    |       |
+------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> describe WORKSHOP;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| Name        | varchar(30) | YES  |     | NULL    |       |
| Date        | date        | YES  |     | NULL    |       |
| Location    | varchar(20) | YES  |     | NULL    |       |
| Facilitator | varchar(30) | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> describe FACILITATOR;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| Name              | varchar(30) | NO   | PRI | NULL    |       |
| sex               | char(1)     | YES  |     | NULL    |       |
| Quilification     | varchar(20) | YES  |     | NULL    |       |
| nationality       | varchar(30) | YES  |     | NULL    |       |
| DateofAppointment | date        | YES  |     | NULL    |       |
| Natureofserivce   | varchar(30) | YES  |     | NULL    |       |
| Salary            | int         | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
7 rows in set (0.00 sec)
```

Above tables are the real tables in the database.

```
mysql> show tables;
+---------------+
| Tables_in_ECCD |
+---------------+
| CHILD          |
| Classroom1     |
| Classroom2     |
| Classroom3     |
| FACILITATOR    |
| GUARDIAN       |
| WORKSHOP       |
+---------------+
7 rows in set (0.01 sec)

mysql> describe Classroom1;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| Id        | int unsigned | NO   |     | 0       |       |
| name      | varchar(30)  | YES  |     | NULL    |       |
| sex       | char(1)      | YES  |     | NULL    |       |
| DOB       | date         | YES  |     | NULL    |       |
| Dzongkhag | varchar(20)  | YES  |     | NULL    |       |
| Geog      | varchar(20)  | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> describe Classroom2;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| Id        | int unsigned | NO   |     | 0       |       |
| name      | varchar(30)  | YES  |     | NULL    |       |
| sex       | char(1)      | YES  |     | NULL    |       |
| DOB       | date         | YES  |     | NULL    |       |
| Dzongkhag | varchar(20)  | YES  |     | NULL    |       |
| Geog      | varchar(20)  | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)

mysql> describe Classroom3;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| Id        | int unsigned | NO   |     | 0       |       |
| name      | varchar(30)  | YES  |     | NULL    |       |
| sex       | char(1)      | YES  |     | NULL    |       |
| DOB       | date         | YES  |     | NULL    |       |
| Dzongkhag | varchar(20)  | YES  |     | NULL    |       |
| Geog      | varchar(20)  | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```

Above colours are the views made from the child table.

3.2 Python Database Connectivity

```
1  from tkinter import *
2  from tkinter import ttk
3  import pymysql
4  from tkinter import messagebox
5
6
7  class Student:
8      def __init__(self, root):
9          self.root = root
10         self.root.title("Student Management System")
11         root.geometry("1350x900+0+0")
12
13         title = Label(self.root, text="Gyalpozhing ECCD", bd=10, relief=GROOVE,
               font=("times new roman", 40, "bold"), bg="cyan", fg="black")
14         title.pack(side=TOP, fill=X)
15
16         # require data variables
17         self.roll_var = StringVar()
18         self.name_var = StringVar()
19         self.gender_var = StringVar()
20         self.dob_var = StringVar()
```

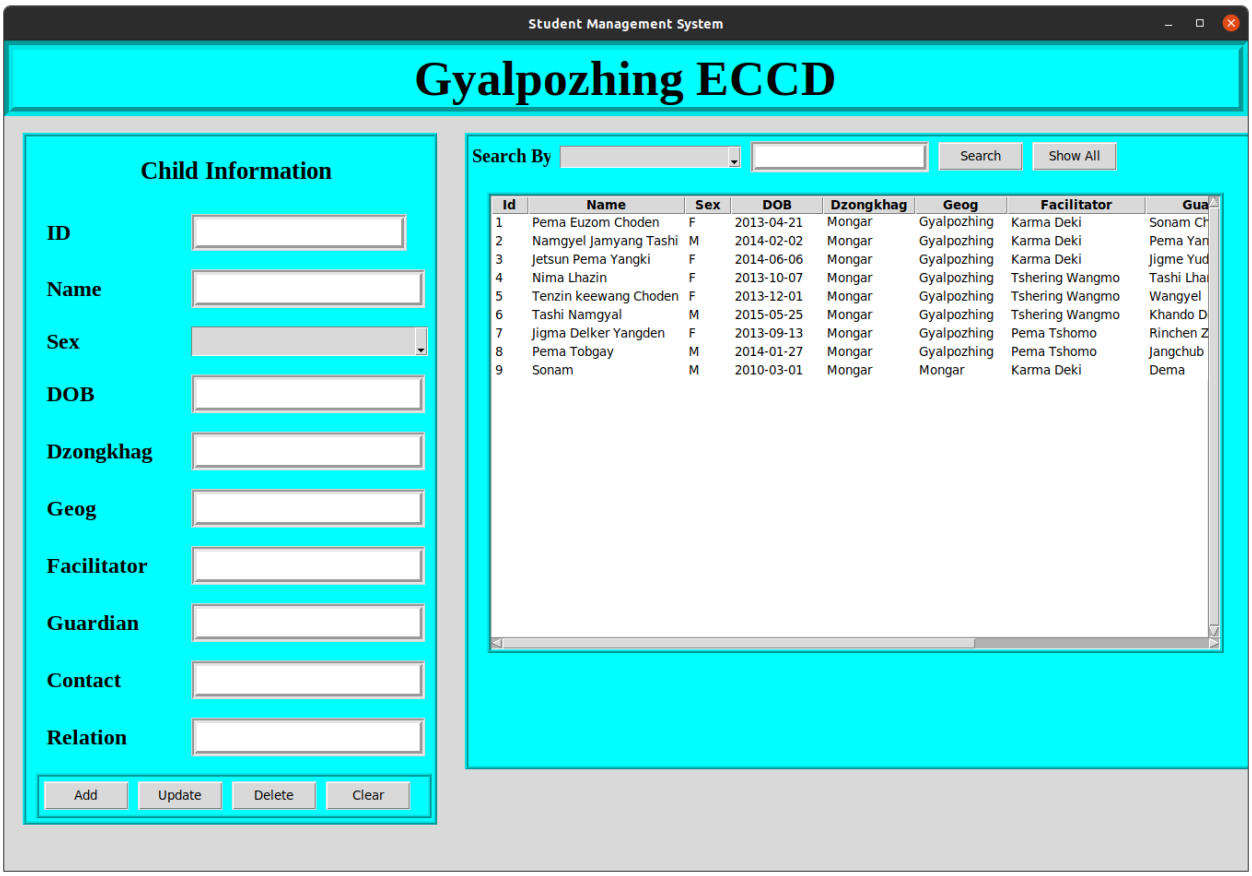
```
21          self.dzong_var = StringVar()
22          self.geog_var = StringVar()
23          self.facilitator_var = StringVar()
24
25          self.guard_var = StringVar()
26          self.contact_var = StringVar()
27          self.relation_var = StringVar()
28
29          self.search_by = StringVar()
30          self.search_txt = StringVar()
31
32          # manage frame left side
33          Manage_Frame = Frame(self.root, bd=4, relief=RIDGE, bg="cyan")
34          Manage_Frame.place(x=20, y=100, width=450, height=750)
35
36          m_title = Label(Manage_Frame, text="Child Information", bg="cyan", fg="
                black",
37                          font=("times new roman", 20, "bold"))
38          m_title.grid(row=0, columnspan=2, pady=20)
39
40          lbl_roll = Label(Manage_Frame, text="ID", bg="cyan", fg="black", font=(
                "times new roman", 18, "bold"))
41          lbl_roll.grid(row=1, column=0, padx=20, pady=10, sticky="w")
42          txt_roll = Entry(Manage_Frame, textvariable=self.roll_var, font=("times
                 new roman", 17, "bold"), bd=5, relief=GROOVE)
43          txt_roll.grid(row=1, column=1, padx=20, pady=10, sticky="w")
44
45          lbl_name = Label(Manage_Frame, text="Name", bg="cyan", fg="black", font
                =("times new roman", 18, "bold"))
46          lbl_name.grid(row=2, column=0, padx=20, pady=10, sticky="w")
47          txt_name = Entry(Manage_Frame, textvariable=self.name_var, font=("times
                 new roman", 18, "bold"), bd=5, relief=GROOVE)
48          txt_name.grid(row=2, column=1, padx=20, pady=10, sticky="w")
49
50          lbl_gender = Label(Manage_Frame, text="Sex", bg="cyan", fg="black",
                font=("times new roman", 18, "bold"))
51          lbl_gender.grid(row=3, column=0, padx=20, pady=10, sticky="w")
52          combo_gender = ttk.Combobox(Manage_Frame, textvariable=self.gender_var,
                 font=("times new roman", 18, "bold"),state="readonly")
53          combo_gender['values'] = ("M", "F")
54          combo_gender.grid(row=3, column=1, padx=20, pady=10, sticky="w")
55
56          lbl_dob = Label(Manage_Frame, text="DOB", bg="cyan", fg="black", font=(
                "times new roman", 18, "bold"))
57          lbl_dob.grid(row=4, column=0, padx=20, pady=10, sticky="w")
58          txt_gender = Entry(Manage_Frame, textvariable=self.dob_var, font=("
                times new roman", 18, "bold"), bd=5, relief=GROOVE)
59          txt_gender.grid(row=4, column=1, padx=20, pady=10, sticky="w")
60
61          lbl_dzong = Label(Manage_Frame, text="Dzongkhag", bg="cyan", fg="black"
                , font=("times new roman", 18, "bold"))
62          lbl_dzong.grid(row=5, column=0, padx=20, pady=10, sticky="w")
63          txt_dzong = Entry(Manage_Frame, textvariable=self.dzong_var, font=("
                times new roman", 18, "bold"), bd=5, relief=GROOVE)
64          txt_dzong.grid(row=5, column=1, padx=20, pady=10, sticky="w")
65
66          lbl_geog = Label(Manage_Frame, text="Geog", bg="cyan", fg="black", font
                =("times new roman", 18, "bold"))
67          lbl_geog.grid(row=6, column=0, padx=20, pady=10, sticky="w")
68          txt_geog = Entry(Manage_Frame, textvariable=self.geog_var, font=("times
                 new roman", 18, "bold"), bd=5, relief=GROOVE)
69          txt_geog.grid(row=6, column=1, padx=20, pady=10, sticky="w")
70
71          lbl_facilitator = Label(Manage_Frame, text="Facilitator", bg="cyan", fg
                ="black",font=("times new roman", 18, "bold"))
72          lbl_facilitator.grid(row=7, column=0, padx=20, pady=10, sticky="w")
73          txt_facilitator = Entry(Manage_Frame, textvariable=self.facilitator_var
                , font=("times new roman", 18, "bold"), bd=5, relief=GROOVE)
```

```
74          txt_facilitator.grid(row=7, column=1, padx=20, pady=10, sticky="w")
75
76          lbl_guard = Label(Manage_Frame, text="Guardian", bg="cyan", fg="black",
                font=("times new roman", 18, "bold"))
77          lbl_guard.grid(row=8, column=0, padx=20, pady=10, sticky="w")
78          txt_guard = Entry(Manage_Frame, textvariable=self.guard_var, font=("
                times new roman", 18, "bold"), bd=5, relief=GROOVE)
79          txt_guard.grid(row=8, column=1, padx=20, pady=10, sticky="w")
80
81          lbl_contact = Label(Manage_Frame, text="Contact", bg="cyan", fg="black"
                , font=("times new roman", 18, "bold"))
82          lbl_contact.grid(row=9, column=0, padx=20, pady=10, sticky="w")
83          txt_contact = Entry(Manage_Frame, textvariable=self.contact_var, font=(
                "times new roman", 18, "bold"), bd=5, relief=GROOVE)
84          txt_contact.grid(row=9, column=1, padx=20, pady=10, sticky="w")
85
86          lbl_relation = Label(Manage_Frame, text="Relation", bg="cyan", fg="
                black", font=("times new roman", 18, "bold"))
87          lbl_relation.grid(row=10, column=0, padx=20, pady=10, sticky="w")
88          txt_relation = Entry(Manage_Frame, textvariable=self.relation_var, font
                =("times new roman", 18, "bold"), bd=5, relief=GROOVE)
89          txt_relation.grid(row=10, column=1, padx=20, pady=10, sticky="w")
90
91          # button frame
92          Button_Frame = Frame(Manage_Frame, bd=4, relief=RIDGE, bg="cyan")
93          Button_Frame.place(x=10, y=690, width=430)
94
95          addbtn = Button(Button_Frame, text="Add", width=8, command=self.
                add_student).grid(row=0, column=0, padx=5, pady=5)
96          updatebtn = Button(Button_Frame, text="Update", width=8, command=self.
                update_data).grid(row=0, column=1, padx=5, pady=5)
97          deletebtn = Button(Button_Frame, text="Delete", width=8, command=self.
                delete_data).grid(row=0, column=2, padx=5, pady=5)
98          clearbtn = Button(Button_Frame, text="Clear", width=8, command=self.
                clear).grid(row=0, column=3, padx=5, pady=5)
99
100         # ==============Detail Frame=========================
101         Detail_Frame = Frame(self.root, bd=4, relief=RIDGE, bg="cyan")
102         Detail_Frame.place(x=500, y=100, width=890, height=690)
103
104         lbl_search = Label(Detail_Frame, width=8, text="Search By", bg="cyan",
                fg="black",
105                            font=("times new roman", 15, "bold"))
106         lbl_search.grid(row=0, column=0, padx=5, pady=5, sticky="w")
107         combo_search = ttk.Combobox(Detail_Frame, textvariable=self.search_by,
                font=("times new roman", 13),
108                                     state="readonly")
109         combo_search['values'] = ("Roll_no")
110         combo_search.grid(row=0, column=1, padx=5, pady=5, sticky="w")
111
112         txt_search = Entry(Detail_Frame, textvariable=self.search_txt, font=("
                times new roman", 13), bd=5,
113                            relief=GROOVE)
114         txt_search.grid(row=0, column=2, padx=5, pady=5, sticky="w")
115
116         searchbtn = Button(Detail_Frame, text="Search", width=8, command=self.
                search_data).grid(row=0, column=3, padx=5, pady=5)
117         showallbtn = Button(Detail_Frame, text="Show All", width=8, command=
                self.fetch_data).grid(row=0, column=4, padx=5, pady=5)
118
119         # Table Frame
120         Table_Frame = Frame(Detail_Frame, bd=4, relief=RIDGE, bg="cyan")
121         Table_Frame.place(x=20, y=60, width=800, height=500)
122
123         scroll_x = Scrollbar(Table_Frame, orient=HORIZONTAL)
124         scroll_y = Scrollbar(Table_Frame, orient=VERTICAL)
125
126         self.Student_table = ttk.Treeview(Table_Frame, columns=("Id", "name", "
```

```
            sex", "DOB", "Dzongkhag", "Geog", "Facilitator", "Guardian", "
            Contact", "Relation"), xscrollcommand=scroll_x.set, yscrollcommand=
            scroll_y.set)
127         scroll_x.pack(side=BOTTOM, fill=X)
128         scroll_y.pack(side=RIGHT, fill=Y)
129         scroll_x.config(command=self.Student_table.xview)
130         scroll_y.config(command=self.Student_table.yview)
131         self.Student_table.heading("Id", text="Id")
132         self.Student_table.heading("name", text="Name")
133         self.Student_table.heading("sex", text="Sex")
134         self.Student_table.heading("DOB", text="DOB")
135         self.Student_table.heading("Dzongkhag", text="Dzongkhag")
136         self.Student_table.heading("Geog", text="Geog")
137         self.Student_table.heading("Facilitator", text="Facilitator")
138         self.Student_table.heading("Guardian", text="Guardian")
139         self.Student_table.heading("Contact", text="Contact")
140         self.Student_table.heading("Relation", text="Relation")
141
142         self.Student_table['show'] = 'headings'
143
144         # setting up widths of cols
145         self.Student_table.column("Id", width=40)
146         self.Student_table.column("name", width=170)
147         self.Student_table.column("sex", width=50)
148         self.Student_table.column("DOB", width=100)
149         self.Student_table.column("Dzongkhag", width=100)
150         self.Student_table.column("Geog", width=100)
151         self.Student_table.column("Facilitator", width=150)
152         self.Student_table.column("Guardian", width=150)
153         self.Student_table.column("Contact", width=150)
154         self.Student_table.column("Relation", width=150)
155
156         self.Student_table.pack(fill=BOTH, expand=1)
157         self.Student_table.bind("<ButtonRelease-1>", self.get_cursor)
158
159         self.fetch_data()
160
161     def add_student(self):
162
163         if self.roll_var.get() == "" or self.name_var.get() == "":
164             messagebox.showerror("Error", "please fill all the fields!!!")
165
166         else:
167             con = pymysql.connect(host="localhost", user="group1", password="
                @2020Lab3", database="ECCD")
168             cur = con.cursor()
169
170             cur.execute("insert into CHILD values(%s, %s, %s, %s, %s, %s, %s)",
                 (self.roll_var.get(),
171             self.name_var.get(), self.gender_var.get(), self.dob_var.get(),
                self.dzong_var.get(),
172             self.geog_var.get(), self.facilitator_var.get()))
173
174             cur.execute("insert into GUARDIAN values(%s, %s, %s, %s)", (self.
                guard_var.get(), self.contact_var.get(), self.relation_var.get
                (), self.roll_var.get()))
175             con.commit()
176             self.fetch_data()
177             self.clear()
178             con.close()
179             messagebox.showinfo("Successful", "Record has been inserted.")
180
181     def fetch_data(self):
182
183         con = pymysql.connect(host="localhost", user="group1", password="
            @2020Lab3", database="ECCD")
184         cur = con.cursor()
185
```

```
186         cur.execute("select * from CHILD left join GUARDIAN on Id = childId")
187         rows = cur.fetchall()
188         if (len(rows) != 0):
189             self.Student_table.delete(*self.Student_table.get_children())
190             for row in rows:
191                 self.Student_table.insert('', END, values=row)
192 
193             con.commit()
194         con.close()
195 
196     def clear(self):
197         self.roll_var.set("")
198         self.name_var.set("")
199         self.gender_var.set("")
200         self.dob_var.set("")
201         self.dzong_var.set("")
202         self.geog_var.set("")
203         self.facilitator_var.set("")
204         self.guard_var.set("")
205         self.contact_var.set("")
206         self.relation_var.set("")
207 
208     def get_cursor(self, evnt):
209         cursor_row = self.Student_table.focus()
210         content = self.Student_table.item(cursor_row)
211         row = content['values']
212         self.roll_var.set(row[0])
213         self.name_var.set(row[1])
214         self.gender_var.set(row[2])
215         self.dob_var.set(row[3])
216         self.dzong_var.set(row[4])
217         self.geog_var.set(row[5])
218         self.facilitator_var.set(row[6])
219         self.guard_var.set(row[7])
220         self.contact_var.set(row[8])
221         self.relation_var.set(row[9])
222 
223     def update_data(self):
224         if self.roll_var.get() == "" or self.name_var.get() == "":
225             messagebox.showerror("Error", "please fill all the fields!!!")
226         else:
227             con = pymysql.connect(host="localhost", user="group1", password="
                    @2020Lab3", database="ECCD")
228             cur = con.cursor()
229 
230             cur.execute("UPDATE CHILD SET name=%s, sex=%s, DOB=%s, Dzongkhag=%s
                    , Geog=%s, FacilitatorName=%s where Id=%s", (self.name_var.get
                    (), self.gender_var.get(), self.dob_var.get(), self.dzong_var.
                    get(), self.geog_var.get(), self.facilitator_var.get(), self.
                    roll_var.get()))
231             cur.execute("UPDATE GUARDIAN SET Name=%s, Contact_No=%s, Relation=%
                    s where childId=%s", (self.guard_var.get(), self.contact_var.
                    get(), self.relation_var.get(), self.roll_var.get()))
232             con.commit()
233             self.fetch_data()
234             self.clear()
235             con.close()
236             messagebox.showinfo("successful", "Record has been updated.")
237 
238     def delete_data(self):
239         con = pymysql.connect(host="localhost", user="group1", password="
                @2020Lab3", database="ECCD")
240         cur = con.cursor()
241 
242         cur.execute("delete from GUARDIAN where childId=%s", self.roll_var.get
                ())
243         cur.execute("delete from CHILD where Id=%s", self.roll_var.get())
244         con.commit()
```

```
245             con.close()
246             self.fetch_data()
247             self.clear()
248             messagebox.showinfo("successful", "Record has been deleted.")
249 
250         def search_data(self):
251 
252             con = pymysql.connect(host="localhost", user="group1", password="
                    @2020Lab3", database="ECCD")
253             cur = con.cursor()
254 
255             sql = "SELECT * FROM CHILD, GUARDIAN WHERE Id = childId and Id = %s"
256             adr = self.search_txt.get()
257 
258             val = cur.execute(sql, adr)
259             if (not val):
260                 messagebox.showinfo("No", "Not available!")
261 
262             rows = cur.fetchall()
263             if len(rows) != 0:
264                 self.Student_table.delete(*self.Student_table.get_children())
265                 for row in rows:
266                     self.Student_table.insert('', END, values=row)
267 
268                 con.commit()
269             con.close()
270 
271 
272 root = Tk()
273 obj = Student(root)
274 root.mainloop()
```



GUI application to access the database

Conclusion

The ECCD database contains records for children, their guardian, the facilitators, and the workshops attended by the facilitators.
From this project the members of the group has learned how to gather requirements for the database, desigh the logical and physical structure of the relations.
Learned to use python to connect to mysql and create a GUI to access the database.

The python connectivity GUI application currently contains the GUI interaction for adding, updating, deleting records in the child table and the guardian table.
We may further develop the GUI to be able to access and create changes in the facilitator table and the classroom views as well.

Bibliography

[1] Klein, B.(2020) Python Course, url: https://www.python-course.eu/python_tkinter.php

[2] sharptutorial (2020) Student Entry Tkinter, url: https://www.sharptutorial.com/database-with-gui-python/

[3] Chaitanya Singh(2020) Normalization in DBMS: 1NF, 2NF, 3NF and BCNF in Database, url: https://beginnersbook.com/2015/05/normalization-in-dbms/